American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

An Approach to Estimating Cloud Resources Prior to New Business Startup

Md. Mahmudul Hasan[a]*, Md. Shamim Hossain[b]

[a,b]*Pabna University of Science and Technology, Pabna and 6600, Bangladesh*
[a]*Email: mahmudul.cse@pust.ac.bd*
[b]*Email: shamimpust3@gmail.com*

Abstract

One of the challenging issues in cloud computing is to estimate plausible cloud resources earlier to begin a new business. Vendors or cloud partners are often falling in risks by investing money without aware of the resources they can avail against their budget. In this study, we aim to build a two-stage model that guides vendors to specify a budget as a model parameter and obtain the possible cloud resources via a methodical process. In the first stage of the model, we consider and simulate the monthly cost calculator of Amazon EC2 service using Row Echelon Form to know which factors are involved in driving total cost for cloud resources and how much they affect the cost. In the second stage, we follow a reverse process to optimize the factor values using Genetic Algorithms for determining cloud resources. Finally, our proposed system is evaluated based on popular error measurement processes, MAE and MRE, and they show the outcomes are significant with moderate results in few cases.

Keywords: Estimation; Cloud resources; Cost factors; Startup; Budget; Genetic algorithms.

1. Introduction

In recent times, the concept of cloud computing becomes one of the wonderful blessings of modern technologies and platforms for IT entrepreneurs and online service consumers. Cloud vendors, one of the key actors in cloud computing architecture, are leveraging the best out of it and mounting their revenue. Service providers avail their potential platforms and services to the vendors based on end user demands.

----

* Corresponding author.

Vendors or a group of cloud partners analyze markets and implement prospective applications built on the stack of such cloud computing platforms, generate revenues from consumers [1] and eventually grab the potential market. This leads a win-win situation among the cloud actors (service providers, vendors and end users) in a particular business domain.

The key challenge, however, is to fill the lack of proper ideas of investing money in the market so that the risks become less and confirms the growing revenues over the period of time. Every business owner plans with a predefined budget prior to start a business and becomes vendors in terms of cloud computing concept. For example, vendors or a group of cloud partners may implement a SaaS (Software-as-a-Service) application that is built on the stack of PaaS (Platform-as-a-Service) and IaaS (Infrastructure-as-a-Service) services. Analyzing along with the proper service items from PaaS and IaaS, vendors should also consider the associated expense that fits within their budget. On the other side of the coin, it is also inevitable to get ideas of plausible service items that vendors can obtain within their planned budget; assuming vendors have a little knowledge or they are reviewing how much amount of services they could get within a budget they can afford to.

Many emerging cloud platforms such Amazon Elastic Compute Cloud (EC2), Google App Engine, Microsoft Live Mesh, Sun Network.com (Sun Grid), GRIDS Lab Aneka etc. provides their services across the world and thousands of vendors are implementing these services towards earning billion dollars of revenue [2]. To avoid risks, therefore, the vendors should have ideas on the possible cloud resources they can obtain according to their budget prior to invest money. Evaluating cloud resources based on a predefined budget can be assumed to be directly associated with cost measurement from purchased service items. The entire money spent is eventually the summation of costs of individual service items. So, it would be a kind of reverse engineering of identifying plausible cloud resources by building a relationship among the cost and factors or inputs that drive the cost.

In literature, several cost measurement and economic studies for cloud resources have been conducted over the period [3-7]. However, any resource prediction methodology or model has not yet been built from a predefined vendor budget that could provide an insight of available cloud resources within their limit, importantly prior to start their business. In this study, we concentrate on building a two-stage model for evaluating cloud resources based on the predefined vendors' budget. We considered Amazon EC2 service items as factors involved in cost measurement and collected relevant data from the Amazon EC2 simple monthly calculator form [8]. Amazon EC2 is widely used web service that provides scalable computing in the cloud via a simple web interface for configuring and obtaining capacity with minimal dilemma [9]. In the following sections, we present the methodology of building the model, the data collection process, the forward procedure to finalize cost measurement and the reverse engineering way for evaluating the cloud resources. In the end, we validate the model accuracy, and conclude with the limitation of this research and possible future works.

2. Research Methodology

In our proposed system, we build a two-stage model to evaluate the possible cloud resources that vendors can obtain from a given budget. Figure 1 depicts the whole picture of our proposed model. In the first stage, we simulate Amazon EC2 simple monthly calculator form to measure the total cost. Hence, the 11 input items,

American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) (2016) Volume 26, No 3, pp 172-187

shown in Table 1, become the driving factors for the simulated result that are responsible for the final cost; factors are analogous to cloud resources. We build a relationship among the driving factors and determine coefficients from a linear equation consisting of driving factors using Row Echelon Form method.

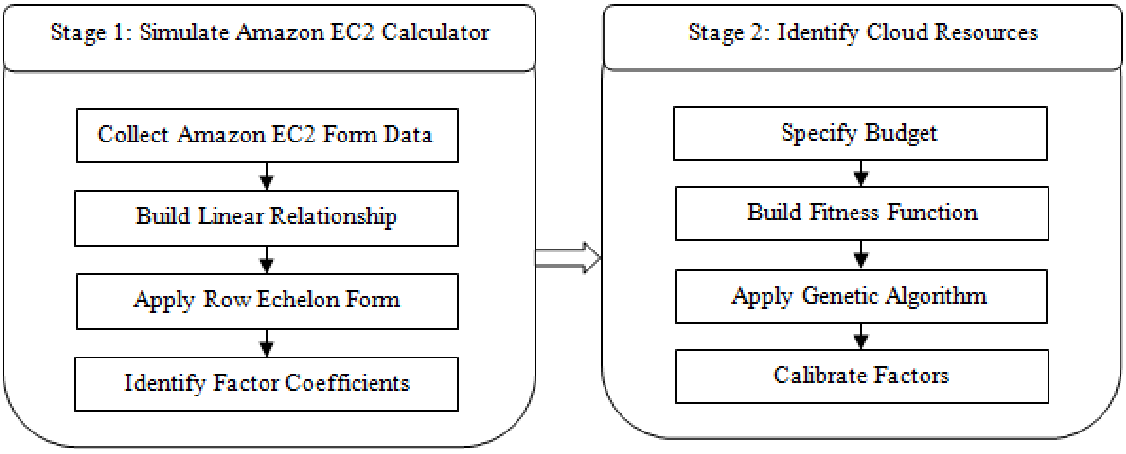

Figure 1: Proposed System

In the second stage of the model, we specify a budget and make an equation with the identified coefficients and factor variables and follow a reverse process for optimizing values of driving factors using Genetic Algorithms that eventually tells us the possible cloud resources the vendors can obtain from the given budget. The given budget in the second stage is ideally the measured cost in the first stage of our model.

Table 1: Driving Factors to the Cost Measurement

Factors	Factors Shorthand
Number of Additional Elastic IPs	NAEIP
Elastic IP Non – attached Time	EIPNAT
Number of Elastic IP Remaps	NEIPR
Inter – Register Data Transfer Out	IRDTO
Data Transfer Out	DTO
Data Transfer In	DTI
VPC Peering Data Transfer	VPCPDT
Intra-Region Data Transfer	IRDT
Public IP/Elastic IP Data Transfer	PEIPDT
Number of Elastic LBs	NELB
Total Data Processed by all ELBs	TDPELB

2.1. Data Collection

Amazon EC2 does not offer direct public data relating to the cost measurement process we deal with in this work. Our proposed model requires analytical data that yields some pattern over the factors involved in the process. We use the simple monthly calculator form for this purpose; we input various range of values in 11 different input fields and observe the total cost from the top tabbed text area that says ‘Estimate of your Monthly Bill ($ x.xx)’. Table 2 insights the range of values for each factor that we use in data collection process. The combination produces a lot of data. We have taken uniform distribution of data for each factor to ensure that the analysis involves every variation of data. Finally, we prepare the data repository using collected data that are alike historical data, preferably used in various cost measurement and prediction systems using Genetic Algorithms [10-12] or/and in regression or linear equation fitting processes [13-14] that build the relationship among the driving variables.

Table 2: Analytical Data Range

Factors	Data Range
NAEIP	1 – 10
EIPNAT	1 - 10 (hours/month)
NEIPR	150 - 350 (/month)
IRDTO	5 – 20 (GB/month)
DTO	15 - 25 (GB/ month)
DTI	2 - 4 (GB/ month)
VPCPDT	2 - 4 (GB/ month)
PEIPDT	5 - 15 (GB/ month)
NELB	3 - 8 (GB/ month)
TDPELB	15 - 65 (GB/ month)

2.2. Initial Cost Measurement

In the data collection process, we accumulate measured costs from the form output where a numerous amount of combinations of factor values have been inputted. The cost variable, *C*, can be defined as the dependent variable and the driving factors as independent variables of a linear relationship. The linear equation can be defined as-

$$C = \sum_{i=1}^{n} a_i x_i \qquad (1)$$

where a_i refers to the coefficient of *i*-th factor, x_i. According to the form output we observe, there are 11 inputs or factors that affect the total cost. However, one among them, IRDT, does not affect significantly to the total cost and so we exclude the factor. Total number of factors, *n* becomes 10. Each combination of factor values constitutes a single linear equation of type *(1)*. An instance of this type is-

$68 = a_1 * 1 + a_2 * 2 + a_3 * 200 + a_4 * 5 + a_5 * 15 + a_6 * 3 + a_7 * 4 + a_8 * 5 + a_9 * 4 + a_{10} * 15$

In this stage, we prepare such equations and aim to determine the coefficients from a_1 to a_{10} using Row Echelon Form which tells how the driving factors affect the total calculated cost. We put the coefficient values into equation *(1)* and conclude the stage one model.

2.3. Calibrating Factors

In the second stage of the model, we use the determined equation of stage one model to optimize the driver factors. In this process, an evolutionary algorithm, the Genetic Algorithms, is used to calibrate the factors. At the end of the process, we obtain possible cloud resources for a budget given as an input to the equation. The model is further validated based on some popular evaluation criteria.

2.4. Model Evaluation

For the validation of the model we use two validation approaches: Mean Absolute Error (MAE) and Mean Relative Error. Both approaches are frequently used in many researches relevant to estimation, prediction and optimization problems.

2.4.1. MAE

Mean Absolute Error (MAE) is very common in evaluating statistical regression based study. Sometimes it was found as better approach over other similar approaches [15]. The common form of MAE is-

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n} \sum_{i=1}^{n} |e_i| \qquad (2)$$

Where n is the number of samples on which the errors are measured. In errors $|e_i| = |f_i - y_i|$, f_i is the measured value and y_i is the original value from the dataset.

2.4.2. MRE

The Magnitude of Relative Error (MRE) is another popular validation process used in statistical and optimization studies. The common form of MRE is-

$$MRE = \frac{Z-Y}{Z} \qquad (3)$$

Where Z is the measured value and Y is the actual value from the dataset.

3. Background Theory

In the proposed model, the coefficients of factors are identified by the Row Echelon Form method. These coefficients represent how each factor controls the total cost. Taking the total cost into account, the Genetic

Algorithms is used to determine the amount of cloud resources can be obtained where cloud resources are eventually the factors values.

3.1. Row Echelon Form

The Row Echelon Form (*ref*) is a form of matrix reduced from an augmented matrix to solve a linear system [16]. The system consists of independent variables from the problem domain. An augmented matrix is a form that represents corresponding coefficients of each variable and constant in the system. A matrix is said to be in *ref* [16],

- If the first nonzero entry in each nonzero row is 1.
- If row k does not consist entirely of zeros, the number of leading zero entries in row $k + 1$ is greater than the number of leading zero entries in row k.
- If there are rows whose entries are all zero, they are below the rows having nonzero entries.

An example of *ref* of an augmented matrix A is,

$$ref\,(A) = \begin{bmatrix} 1 & 0 & 3 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Reduced Row Echelon Form (*rref*) is, a stricter variant of *ref*, a matrix that can be used to solve a linear equation [17]. A matrix is said to be in *rref* if-

- The matrix is in row-echelon form.
- Each leading 1 is the only nonzero entry in its column.

An example of *rref* of an augmented matrix A is,

$$rref\,(A) = \begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matrix can be transformed to its reduced row echelon form, or row reduced to its reduced row echelon form using the elementary row operations. These are:

- Interchange one row of the matrix with another of the matrix.
- Multiply one row of the matrix by a nonzero scalar constant.
- Replace the one row with the one row plus a constant times another row of the matrix.

An example of transforming an augmented matrix to its echelon and reduced echelon form can be illustrated as following.

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 7 & 8 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 2 & 7 & 8 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 3 & 6 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

(A) (A_1) (A_2) (*ref* (A)) (*rref* (A))

To transform matrix A into its echelon forms, we implement the following series of elementary row operations:

Step 1*:* We found the first non-zero entry in the first column of the matrix in row 2; so we interchanged Rows 1 and 2, resulting in matrix A_1.

Step 2*:* Working with matrix A_1, we multiplied each element of Row 1 by -2 and added the result to Row 3 that produced A_2.

Step 3*:* Working with matrix A_2, we multiplied each element of Row 2 by -3 and added the result to Row 3 which produced *ref* (A), the row echelon form.

Step 4*:* Working with matrix *ref* (A), we multiplied the second row by -2 and added it to the first row that produced *rref* (A), the reduced row echelon form.

At the end of the process we get some free and leading variables that yield the solution of the given linear equation [16-17]. In this study, we put the values of driving factors into equation *(1)* and build the linear equations. Later, we find the values of all coefficients from a_1 to a_{10} using Row Echelon Form.

4. Genetic Algorithms

Genetic Algorithm (GA) is a search procedure based on the mechanics of natural selection and genetics [18]. This biologically evolved method is used to solve many constrained and unconstrained optimization problems [19]. The algorithm constantly changes a population of individual solutions. Often the initial population is generated randomly based on the properties of the genes that are responsible for driving the optimization. In a particular generation, the GA randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. A proper selection procedure is applied to the children to pick the fittest children that survive in generations further to reproduce better parents, eventually better children. Over successive generations, the population evolves toward an optimal solution.

In the process of natural evolution, a genetic variation is required for better solution that is analogous to survival of the fittest creatures in the natural world. Selecting better candidates in current generation reproduce more suitable candidates in future generations. Often a genetic diversity amongst solutions prevents local optimum solutions that are not the real optimum. To avail these advantages, three genetic operators: selection, crossover and mutation are used in genetic programming.

Every genetic programming consists of one or multi objective function or fitness function that either applies to maximization or minimization problems. In a given optimization problem, the fitness function provides a fitness value from a relation among function variables or factors that drive fitness function. Many search and optimization problems solved by GA are often involved to a number of constraints which the optimal solution must satisfy and related to the fitness function. The constraints can be expressed as equality, inequality and/or in a range of values among variables related to the optimization problem [20].

Genetic Algorithms has been very popular in diverse problem domains including cloud computing. These include optimizing various task scheduling algorithms for FIFO policy, load balancing in cloud infrastructure, QoS (Quality of Service) aware service composition in cloud computing, scheduling workflow applications in cloud computing environments [21-22] and so on.

5. Model Building

In the first stage of the proposed model, we prepare the augmented matrix using the equations consisting of data produced for each parameter of monthly calculator. Then we generate the matrices, *ref* and *rref*. We prepare the equations using them, and identify the coefficients. In the second stage, we use these coefficients as a base for calculating cloud resources. The fitness function is built using the factors (specified in Table 1) and coefficients obtained from the stage one model.

5.1. Development Environment

The reduced row echelon form in stage one model is determined by a calculator from Linear Algebra Toolkit [23] and implemented in C++. We simulate the genetic algorithm with constrained minimization [24] in MATLAB R2010a using built-in function available for GA.

5.2. First Stage Model

The coefficients, *a1* to *a10*, along with the input value for each factor constitutes many equations. We have taken 10 such equations with varying factor values and the associated costs for them. Equations *(3)* to *(12)* indicate these relationships.

$$a_1 * 1 + a_2 * 2 + a_3 * 200 + a_4 * 5 + a_5 * 15 + a_6 * 3 + a_7 * 4 + a_8 * 5 + a_9 * 4 + a_{10} * 15 = 68 \quad (3)$$

$$a_1 * 1 + a_2 * 2 + a_3 * 200 + a_4 * 5 + a_5 * 25 + a_6 * 3 + a_7 * 2 + a_8 * 5 + a_9 * 3 + a_{10} * 20 = 51 \quad (4)$$

$$a_1 * 1 + a_2 * 2 + a_3 * 200 + a_4 * 5 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 5 + a_9 * 3 + a_{10} * 20 = 51 \quad (5)$$

$$a_1 * 8 + a_2 * 9 + a_3 * 160 + a_4 * 8 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 5 + a_9 * 3 + a_{10} * 20 = 72 \quad (6)$$

$$a_1 * 5 + a_2 * 4 + a_3 * 260 + a_4 * 5 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 5 + a_9 * 3 + a_{10} * 20 = 71 \quad (7)$$

$$a_1 * 7 + a_2 * 3 + a_3 * 300 + a_4 * 9 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 5 + a_9 * 3 + a_{10} * 20 = 82 \quad (8)$$

$$a_1 * 6 + a_2 * 5 + a_3 * 250 + a_4 * 17 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 5 + a_9 * 3 + a_{10} * 20 = 74 \quad (9)$$

$$a_1 * 6 + a_2 * 5 + a_3 * 250 + a_4 * 17 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 10 + a_9 * 4 + a_{10} * 30 = 92 \quad (10)$$

$$a_1 * 6 + a_2 * 5 + a_3 * 250 + a_4 * 17 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 7 + a_9 * 8 + a_{10} * 65 = 166 \quad (11)$$

$$a1 * 6 + a2 * 5 + a_3 * 250 + a_4 * 17 + a_5 * 20 + a_6 * 2 + a_7 * 3 + a_8 * 15 + a_9 * 5 + a_{10} * 50 = 111 \quad (12)$$

From the above equations, we prepare the following the augmented matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 200 & 5 & 15 & 3 & 4 & 5 & 4 & 15 & 68 \\ 1 & 2 & 200 & 5 & 25 & 3 & 2 & 5 & 3 & 20 & 51 \\ 1 & 2 & 200 & 5 & 20 & 2 & 3 & 5 & 3 & 20 & 51 \\ 8 & 9 & 160 & 8 & 20 & 2 & 3 & 5 & 3 & 20 & 72 \\ 5 & 4 & 260 & 5 & 20 & 2 & 3 & 5 & 3 & 20 & 71 \\ 7 & 3 & 300 & 9 & 20 & 2 & 3 & 5 & 3 & 20 & 82 \\ 6 & 5 & 250 & 17 & 20 & 2 & 3 & 5 & 3 & 20 & 74 \\ 6 & 5 & 250 & 17 & 20 & 2 & 3 & 10 & 4 & 30 & 92 \\ 6 & 5 & 250 & 17 & 20 & 2 & 3 & 7 & 8 & 65 & 166 \\ 6 & 5 & 250 & 17 & 20 & 2 & 3 & 15 & 5 & 50 & 111 \end{bmatrix}$$

Using the reduced row echelon calculator [23], after 50 successive row reduced steps, we obtain *rref* (A).

***rref* (A)** =

$$\begin{bmatrix} 1 & 2 & 200 & 5 & 15 & 3 & 4 & 5 & 4 & 15 & 68 \\ 0 & 1 & 1440/7 & 32/7 & 100/7 & 22/7 & 29/7 & 5 & 29/7 & 100/7 & 472/7 \\ 0 & 0 & 1 & 13/865 & 43/692 & 41/3460 & 11/692 & 7/346 & 11/692 & 43/692 & 949/3460 \\ 0 & 0 & 0 & 1 & -10/589 & 5/19 & 19/62 & 255/1178 & 19/62 & -10/589 & 4977/1178 \\ 0 & 0 & 0 & 0 & 1 & -1/5 & -1/5 & 0 & -1/5 & 1 & -17/5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1/2 & -5/2 & 17/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5/7 & 4/7 & 15/7 & 8482/1337 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/5 & 2 & 18/5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 205/23 & 424/23 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/10 \end{bmatrix}$$

According to the process, we generate the following equations from the above matrix that eventually determine the coefficients presented in Table 3.

$$a_1*1 + a_2*2 + a_3*200 + a_4 * 5 + a_5 * 15 + a_6 * 3 + a_7 * 4 + a_8 * 5 + a_9 * 4 + a_{10} * 15 = 68$$

$$a_2*1 + a_3*(1440/7) + a_4*(32/7) + a_5*(100/7) + a_6*(22/7) + a_7*(29/7) + a_8*5 + a_9*(29/7) + a_{10}*(100/7) = 472/7$$

$$a_3 * 1 + a_4 * (13/865) + a_5 * (43/692) + a_6 * (41/3460) + a_7 * (11/692) + a_8 * (7/346) + a_9 * (11/692) + a_{10} * (43/692) = 949/3460$$

$$a_4 * 1 - a_5 * (10/589) + a_6 * (5/9) + a_7 * (19/62) + a_8 * (255/1178) + a_9 * (19/62) - a_{10} * (10/589) = 4977/1178$$

$a_5 * 1 - a_6 * (1/5) - a_7 * (1/5) - a_9 * (1/5) + a_{10} * 1 = -17/5$

$a_6 * 1 + a_9 * (1/2) - a_{10} * (5/2) = 17/2$

$a_7 * 1 + a_8 * (5/7) + a_9 * (4/7) + a_{10} * (15/7) = 8482/1337$

$a_8 * 1 + a_9 * (1/5) + a_{10} * 2 = 18/5$

$a_9 * 1 + a_{10} * (205/23) = 424/23$

$a_{10} = 1/10$

Table 3: Coefficients from Stage One Model

Coefficients	Value
a_1	3.482
a_2	0.068
a_3	0.099
a_4	0.037
a_5	-0.759
a_6	-0.022
a_7	-3.817
a_8	-0.109
a_9	17.544
a_{10}	0.1

The coefficients are now used to prepare the fitness function for Genetic Algorithms in the second stage of the model.

5.3. Second Stage Model

Building fitness function involves 10 independent variables that are driving factors for the measured cost in stage one; measured cost is the dependent variable. The coefficient values indicate how they drive the measured cost with their associating factors together. In this stage, we put the factors together with coefficient values to build the fitness function. We use the built-in function for genetic algorithm in Matlab for the analysis and we require 10 GA variables ranging from *x(1)* to *x(10)*, representing 10 factors, and *y* for cost. The fitness function, therefore, becomes-

$$y = 3.482 * x(1) + 0.068 * x(2) + 0.099 * x(3) + 0.037 * x(4) - 0.759 * x(5) - 0.022 * x(6) - 3.817 * x(7) - 0.109 * x(8) + 17.544 * x(9) + 0.1 * x(10) \qquad (13)$$

The constraints of GA in this study are a range of values indicated in Table 2. The reason behind this is we choose the range to restrict the solution inside the perimeter of the values taken for the analysis. The commands and code snippets for the analysis are the following.

Fitness Function:

function y = objective_fitness (x)

y = 3.482 * *x(1)* + 0.068 * *x(2)* + 0.099 * *x(3)* + 0.037 * *x(4)* - 0.759 * *x(5)* - 0.022 * *x(6)* - 3.817 * *x(7)* - 0.109 * *x(8) + 17.544 * x(9) + 0.1 * x(10);*

end

Constraints:

function [c, ceq] = factor_ constraints (x)

c = [- x (1) + 1; x (1) - 10; - x (2) + 1; x (2) - 10; -x (3) + 150; x (3) - 350;

- x (4) + 5; x (4) - 20; -x (5) + 15; x (5) - 25; -x (6) + 2; x (6) - 4;

-x (7) + 2; x (7) - 4; -x (8) + 5; x (8) - 15; -x (9) + 3; x (9) - 8;

-x (10) + 15; x (10) - 65;

] ;

ceq = [];

end

Genetic Options:

options = gaoptimset ('PopulationSize', 100, 'Generations', 50, 'CrossoverFcn', @crossoversinglepoint, 'CrossoverFraction', 0.8, 'MutationFcn', {@mutationgaussian, 0.2});

Genetic Algorithm:

[x, fval] = ga(@objective_fitness, 10, [],[],[],[],[],[], @ factor_ constraints, options)

It can be noted from the above code snippets that a built in function *ga()* has been used to calculate the factor values in *x* based on fitness value, *fval* [24]. The first parameter is the function that defines the fitness function. The second parameter is the number of factors involved in the proposed optimization problem. The final two parameters represent the constraint function that taken ranges of values from Table 2 and the options for genetic operators, respectively. The other parameters are optional. It can also be noted that the variable *options* holds the configuration of the genetic operators. The chosen values for each set of values are considered as a standard for Genetic Algorithms. For the given settings and code snippet we obtain values for *x(1)* to *x(10)* which are eventually the optimized factor values that are analogous to cloud resources that this study aims to determine for. In the following section, we represent our experimental results, system evaluation and limitation of this work.

6. Results and Discussion

The cloud resources are obtained as a form of model variables from *x(1)* to *x(10)* that represent the factors in Table 1, from top to bottom respectively. That means, *x(1)* corresponds to Number of Additional Elastic IPs (NAEIP), *x(2)* to Elastic IP Non – attached Time (EIPNAT) and so *x(10)* to Total Data Processed by all ELBs (TDPELB). We specify a given budget in the fitness cost function and obtain the cloud resources using the GA. Table 4 presents the amount of cloud resources we obtain for a set of few cases of given budgets.

It is noted that, the cloud resources in Table 4 are all in rounded values of the original fractional values that GA analysis generates as not all factors can accept fractional values; for example: NAEIP. Across all cases, the variation of the amount of resources for each factor is minor except the variable *x(3)* where the amount of resources have increased with the rise of the given budget. It can be inferred that the factor NEIPR mostly affects the given budget according to the analysis on data obtained from the monthly calculator. Taken as a whole, the proposed system provides a way to get an idea of cloud resources obtainable from a given budget.

Table 4: Amount of Cloud Resources Obtained from Given Budgets

Case No.	Given Budget	*x(1)*	*x(2)*	*x(3)*	*x(4)*	*x(5)*	*x(6)*	*x(7)*	*x(8)*	*x(9)*	*x(10)*
1	45	3	2	266	15	25	15	3	4	2	19
2	50	1	2	206	6	17	3	3	9	3	20
3	56	2	2	206	7	17	2	3	5	3	21
4	69	2	5	169	7	19	3	3	7	4	22
6	89	4	2	322	6	25	3	4	9	4	24

The proposed system is evaluated on the basis of error measurement by Mean Absolute Error (MAE), Mean Relative Error (MRE). Once we put factor values generated by the second stage model along with the coefficients determined by first stage model, we obtain the measured cost by our proposed model. Table 5 shows the error rates measured against original given budget (GB). C_{AWS} refers to the AWS calculator generated cost when we put corresponding input values taken from Table 4. Similarly, C_{PS} shows the cost generated by the object function once we put corresponding values from Table 4. AE_{GBCPS} and RE_{GBCPS} present the absolute error and relative error respectively for error measurement between the given budget and cost measured by the proposed system. Similarly, $AE_{CAWSCPS}$ and $RE_{CAWSCPS}$ refer to the absolute error and relative error respectively for error measurement between the amazon monthly calculated cost and cost measured by the proposed system.

Table 5: Absolute and Relative Errors of Proposed System

Case No.	GB	C_{AWS}	C_{PS}	AE_{GBCPS}	RE_{GBCPS}	$AE_{CAWSCPS}$	$RE_{CAWSCPS}$
1	45	46.75	43.27	1.73	0.038	3.48	0.074
2	50	50.88	53.47	3.47	0.069	2.59	0.051
3	56	54.35	55.14	0.86	0.015	0.79	0.015
4	69	71.11	69.97	0.97	0.014	1.14	0.016
5	89	92.53	83.45	5.55	0.062	9.08	0.098

From the error analysis, we observe that the proposed model works pretty well on the basis of MRE (0.05) over 5 samples though the MAE (3.42) value is slightly higher. The MAE and MRE values are calculated based on the measured cost by our system against the AWS generated cost. The MAE (2.52) and MRE (0.04) values for the given budget against the cost measured by the proposed system works in acceptable range. It is also noted that, the AWS generated cost (C_{AWS}) for our identified cloud resources are too close to the proposed system measured cost (C_{PS}). That verifies that our simulation to AWS monthly calculator can be acceptable.

7. Limitations and Recommendations

The performance of the proposed system works very well in some extents and somewhat moderate in few cases. The absolute errors and relative errors suggest that the model requires more fine tuning of the coefficients of the driving factors. We analyze very limited number of sample data in our study which may lead to some false cases where our model may not work well. Also, the Row Echelon Form can take much more time for larger set of data as it has less capacity of calculating many equations. So, a more dynamic way can be considered by implementing a bot program for extracting real time input values for better historical data and a better algorithm or process such as Artificial Neural Network (ANN) can assist in further improvement of the model. Nevertheless, the proposed model works better than the estimated budget given as an input to the algorithm we proposed. We recommend this model as a basis for other approaches taken on other platforms such as Microsoft Azure and similar. Developers can build tools using the coefficients of the factors and vendors obtain the idea of resources straight away. Also researchers can utilize the concept of model improvement in stepwise stages that may lead to further improvement in any cost measurement model.

8. Conclusion and Future Works

Estimating cloud resources is beneficial prior to open a new business on stack of cloud computing. It is also challenging for cloud partners or vendors to invest money for services that they are not completely knowledgeable or aware of. In this study, we aim to implement a system that allow vendors to specify a budget and obtain plausible cloud resources that guide them to decide which cloud services in what amount they can consume prior starting their own services within their budget. For this purpose, we refer to Amazon EC2 monthly cost calculator to collect a set of data of inputs or factors that drive the total cost. We simulate the cost measurement process to identify the contribution of factors that how they affect the total cost. We determine the coefficient of each factor using Row Echelon Form as we solve the cost measurement process as linear equation. Later, we apply a reverse process of optimizing factor values using the obtained coefficients and historical data taken from the Amazon EC2 calculator. An evolutionary algorithm, GA, is used to solve this optimization problem and finally a set of cloud resources are determined against a given budget.

The performance of the proposed model is reasonably well based on the limited number of data as shows by Mean Absolute Error and Mean Relative Error. It can be further improved by applying other machine learning algorithms such as Artificial Neural Network (ANN) as the historical data lies in various ranges of values. Also, the cost measurement process can be implemented in real time by collecting data using automated programs. We aim to accommodate these tasks as future works.

References

[1] L. J. Zhang and Q. Zhou. “CCOA: Cloud computing open architecture,” in IEEE International Conference on Web Services, 2009, pp. 607-616.

[2] R. Buyya, C.S. Yeo, and S. Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in 10th IEEE International Conference on High Performance Computing and Communications, 2008, pp. 5 - 13.

[3] R Buyya, H. Stockinger, J. Giddy, D. Abramson. “Economic models for resource management and scheduling in grid computing,” Concurrency and Computation: Practice and Experience, vol. 14, pp. 1507-1542, Dec. 2002.

[4] G.A. Thanos, C. Courcoubetis, G.D. Stamoulis. “Adopting the Grid for Business Purposes: The Main Objectives and the Associated Economic Issues,” in International Workshop on Grid Economics and Business Models, 2007, pp. 1 - 15.

[5] J. Hwang, J. Park. “Decision Factors of Enterprises for Adopting Grid Computing,” in International Workshop on Grid Economics and Business Models, 2007, pp. 16 – 28.

[6] J. Gray. “Distributed Computing Economics,” in Queue 6.3, 2008, pp. 63-68.

[7] D. Kondo, B. Javadi, P. Malecot. “Cost-Benefit Analysis of Cloud Computing versus Desktop Grids,” in IEEE International Symposium on Parallel & Distributed Processing, 2009, pp. 1 - 12.

[8] “Amazon EC2 simple monthly calculator.” Internet:

https://calculator.s3.amazonaws.com/index.html, [Sep. 13, 2016].

[9] “Amazon Elastic Compute Cloud.” Internet: https://aws.amazon.com/ec2/, [Sep. 13, 2016].

[10] P. Reed, B.S. Minsker, D.E. Goldberg. “A multiobjective approach to cost effective long-term

groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data," Journal of Hydroinformatics, vol. 3, pp. 71-89, 2002.

[11] P.F. Pai, W.C. Hong. "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," Electric Power Systems Research, vol. 74, pp. 417-425, Jun. 2005.

[12] Y. Gao, H. Rong, J.Z. Huang. "Adaptive grid job scheduling with genetic algorithms," Future Generation Computer Systems, vol. 21, pp. 151-161, Jan. 2005.

[13] H.K.N. Leung, L. White. "A cost model to compare regression test strategies," in Proc. IEEE Conference on Software Maintenance, 1991, pp. 201-208.

[14] M.R. Young, R. Martin. "A minimax portfolio selection rule with linear programming solution," Management Science, vol. 44, pp. 673-683, 1998.

[15] T. Chai, R.R. Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)?– Arguments against avoiding RMSE in the literature," Geoscientific Model Development, vol. 7, pp. 1247-1250, 2014.

[16] S. J. Leon. (1980). Linear algebra with applications. (8th edition). [On-line]. Available: http://www.academia.edu/download/36340306/Linear_Algebra_with_Applications_-_8th_-_Leon.pdf [Aug 05, 2016].

[17] "Linear Algebra / Row Reduction and Echelon Forms." Internet: https://en.wikibooks.org/wiki/Linear_Algebra/Row_Reduction_and_Echelon_Forms, [Aug. 05, 2016].

[18] D. E. Goldberg. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. (1st edition).

[19] "Genetic Algorithm." Internet: https://www.mathworks.com/discovery/genetic-algorithm.html, [Aug. 13, 2016].

[20] K. Deb. "An efficient constraint handling method for genetic algorithms," Computer Methods in Applied Mechanics and Engineering, vol. 186, pp. 311-338, 2000.

[21] Y. Ge, G. Wei. "GA-based task scheduler for the cloud computing systems," in IEEE International Conference on Web Information Systems and Mining (WISM), 2010, pp. 181-186.

[22] K. Dasgupta, B. Mandal, P. Dutta, et al. "A genetic algorithm (ga) based load balancing strategy for cloud computing," in International Conference on Computational Intelligence Modeling Techniques and Applications, 2013, pp. 340-347.

[23] "Find the matrix in reduced row echelon form - Linear Algebra Toolkit." Internet:

www.math.odu.edu/~bogacki/cgi-bin/lat.cgi?c=rref, [Aug. 6, 2016].

[24] "Constrained Minimization Using the Genetic Algorithm." Internet: https://www.mathworks.com/help/gads/examples/constrained-minimization-using-the-genetic-algorithm.html, [Aug. 15, 2016].